interweb

Generated on 15/11/2026 00:12

KNOWLEDGE BASE · PUBLIC DOCUMENT

Updating a Docker application without losing data

Docker & Containers

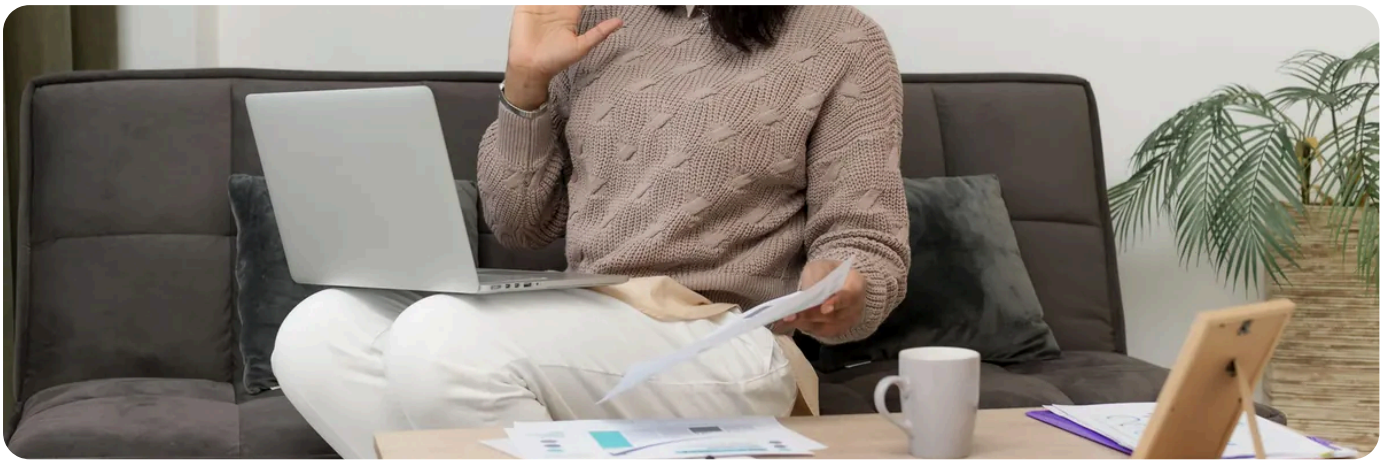
CODE	VERSION	COUNTRY · LANGUAGE
RLX-AO-KB-870	v2026.10.10	Angola · English

Interweb Angola
RLX Angola, Lda. · Group RLX Holdings
Tax ID (NIF) 500 260 5724 · Commercial registration no. 26464-25/250724

https://www.meu.interweb.ao/knowledgebase/870

Updating a Docker application without losing data

Docker & Containers Created on **October 10, 2026** 3 min read



Updating an application in Docker is, as a rule, pulling the new image and recreating the container. The data survives **if it lives in a volume**, and that is where everything is won or lost: the container is disposable, the volume is not. With a backup made first and a pinned version, updating is a calm procedure that you can undo. This article assumes a VPS with Docker and a Compose project (**example**).

The procedure

1. **Read the notes for the new version.** The project publishes what changed and what needs extra steps (migrations, renamed variables). Jumping several versions at once is where problems tend to appear.

2. **Dump the database:**

```
docker compose exec -T db pg_dump -U appuser appdb > backup.sql
```

The `-T` is needed so the `>` does not receive junk from a terminal. For MariaDB or MySQL the equivalent is `mysqldump`. See **backups and restores**.

3. **Keep the configuration file and the `.env` too** somewhere off the server. Copying a volume and the whole VPS are covered in **backing up a VPS**.

4. **Pin the version you are moving to** in the file (`image: name:1.2.3` rather than `:latest`). That way you know exactly what you installed and can go back.

5. **Pull and recreate:**

```
docker compose pull
docker compose up -d
```

Compose only recreates the services whose image changed. The volumes stay.

6 **Read the logs** and check the application started and the migrations ran: `docker compose logs -f --tail=100`. Open the application and try a real operation.

7 **Clean up old images** only when you are happy: `docker image prune`. Until you do, going back is just changing the number and repeating step 5.

What deletes data and what does not

Command	What it does	The data?
`docker compose pull` + `up -d`	Pulls images and recreates the containers that changed.	Stays, if it is in volumes.
`docker compose down`	Stops and removes the containers and the network.	Stays. Volumes are not touched.
`docker compose down -v`	Does the same and removes the volumes too.	**Lost.**
`docker volume prune` and `docker system prune --volumes`	Remove volumes that no container is using (depending on version and options, only anonymous ones or named ones too).	**Lost**, if the containers are stopped or removed.
Files written outside a volume	They live in the container layer.	**Lost** when the container is recreated.

An "unused" volume is a volume with no running container.
After a `docker compose down`, your database volume has no container attached, and a `prune` run at that moment can take it. Before any `prune` that includes volumes, look at the list with `docker volume ls` and have a backup. Otherwise, do not use `--volumes`.

A database's major version is not changed by changing the number.
Going, say, from one major version of PostgreSQL to the next needs a dump (`pg_dump`) and a restore into a new volume. For minor updates within the same major version, the procedure above is enough.

Rehearse with a copy.
If the application matters, restore the backup into a new folder with another project name, update there, and only then touch what is live. And note: updating Docker and the application is yours on an unmanaged VPS: **how far our support goes**.

NEED A HAND?

Something went wrong during the update and the server stopped answering? If it is the VPS, the network or access, we deal with it; tell us what you saw.

Open a support ticket →

SEE ALSO

- Backing up with mysqldump and pg_dump, and restoring the copy →
- Backing up a VPS: snapshots or files →
- Docker filling the disk: images, volumes and logs →

interweb

Interweb Angola

RLX Angola, Lda. · Group RLX Holdings

TAX ID (NIF)	500 260 5724
COMMERCIAL REGISTRATION NO.	26464-25/250724
REGISTERED OFFICE	Rua Direita do Patriota, Bairro Dangereux Belas, Luanda, Angola
CONTACT	info@interweb.ao · +244 942 011 191 · www.interweb.ao
OPENING HOURS	Monday to Friday - 08 AM to 04 PM Saturday & Sunday - Closed

Online version of this document

RLX-AO-KB-870 · v2026.10.10 · Generated on 15/11/2026 00:12